Focus-dLLM: Accelerating Long-Context Diffusion LLM Inference via Confidence-Guided Context Focusing

Lingkun Long[1,3], **Yushi Huang**[2,3], **Shihao Bai**[3], **Ruihao Gong**[1,3], **Jun Zhang**[2], **Ao Zhou**[1], **Jianlei Yang**[1*]
[1]Beihang University, [2]Hong Kong University of Science and Technology, [3]SenseTime Research
Email: {longxmas, aozhou, jianlei}@buaa.edu.cn

Abstract

Diffusion Large Language Models (dLLMs) deliver strong long-context processing capability in a non-autoregressive decoding paradigm. However, the considerable computational cost of bidirectional full attention limits the inference efficiency. Although sparse attention is promising, existing methods remain ineffective. This stems from the need to estimate attention importance for tokens yet to be decoded, while the unmasked token positions are unknown during diffusion. In this paper, we present **Focus-dLLM**, a novel training-free attention sparsification framework tailored for accurate and efficient long-context dLLM inference. Based on the finding that token confidence strongly correlates across adjacent steps, we first design a *past confidence-guided indicator* to predict unmasked regions. Built upon this, we propose a *sink-aware pruning strategy* to accurately estimate and remove redundant attention computation, while preserving highly influential attention sinks. To further reduce overhead, this strategy reuses identified sink locations across layers, leveraging the observed cross-layer consistency. Experimental results show that our method offers more than $29\times$ lossless speedup under $32K$ context length. The code is publicly available at: https://github.com/Longxmas/Focus-dLLM.

1 Introduction

Diffusion large language models (dLLMs) (Bie et al., 2025; Gong et al., 2025; Arriola et al., 2025) have recently emerged as a compelling non-autoregressive paradigm for text generation, replacing left-to-right token emission with iterative denoising over a fixed-length sequence (Li et al., 2022; Gong et al., 2022; Austin et al., 2021; Lou et al., 2023; He et al., 2023). By updating multiple positions in parallel and leveraging bidirectional attention, dLLMs offer an appealing path toward higher decoding throughput while retaining strong generation quality. Moreover, recent studies have substantially extended the context length of dLLMs (Liu et al., 2025a; He et al., 2025), demonstrating effective long-context extrapolation and scaling to long inputs.

Nevertheless, efficient long-context inference remains a key obstacle for the dLLM due to its *non-autoregressive* decoding and bidirectional *full* attention nature. Prior methods (Wu et al., 2025; Liu et al., 2025b; Ma et al., 2025) to address this challenge fall into two categories: (*i*) *Approximated* KV cache and (*ii*) sparse attention. The former selectively refreshes KV states by exploiting strong redundancy between adjacent steps. However, attention computation is still costly over the *full* cached context. On the other hand, sparse attention (Tang et al., 2024; Xiao et al., 2024a; Xu et al., 2025; Yuan et al., 2025) offers a practical solution, but it often requires token importance estimation using the *currently decoded* token as a query (Zhang et al., 2023; Xiao et al., 2024a). Since the positions to be decoded (unmasked) are not known in advance for dLLMs, recent works (Song et al., 2025; Huang et al., 2025a) leverage inaccurate coarse-grained estimation, leading to suboptimal performance and limited efficiency. This paper, therefore, asks: *Can we accurately predict the positions of the unmasked tokens and only retain necessary computation to achieve more effective long-context inference acceleration for dLLMs?*

To tackle this challenge, we first make an in-depth analysis to investigate the predictability of the unmasked tokens. In particular, we discover that the confidence scores at the same positions in two consecutive steps exhibit a strong positive correlation, and the positions of currently unknown tokens largely overlap with those that had the highest-confidence tokens in the previous step. Thus, unmasked positions for the current steps can be inferred from previous-step confidence. Besides, we

*Jianlei Yang is the corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 12146–12160

also analyze the redundancy of attention patterns and observe that attention sink (Xiao et al., 2023; Ruscio et al., 2025), which contributes significantly to the attention score in LLMs (Bai et al., 2023; Touvron et al., 2023), displays notable cross-layer consistency for dLLMs. This phenomenon suggests sink tokens can be identified at an intermediate depth. Therefore, we can directly reuse them without re-identification in deeper layers.

Motivated by the above findings, we propose Focus-dLLM, a training-free sparse attention framework with approximated KV cache, to accelerate long-context dLLM inference. To begin with, we introduce a *past confidence-guided indicator* that uses confidence scores from step $t-1$ to predict the unmasked positions at step t, and then window-expands them to preserve semantic coherence. Next, we design a *sink-aware pruning strategy* for diffusion decoding: Using the tokens within the positions predicted before as queries, we select only the most relevant tokens for attention while retaining step-wise attention sinks. Moreover, this approach shares the identified sink tokens across layers to further reduce additional overhead. Leveraging these novel techniques, our framework computes attention over the predicted unmasked queries and the selected necessary key-value pairs. As a result, it achieves considerable inference speedups (Huang et al., 2025c, 2024; Zhu et al., 2026) without compromising performance throughout the dynamic decoding process.

Our contributions are summarized as follows:

- We analyze diffusion inference dynamics and reveal a strong positive correlation of token confidence across adjacent denoising steps, together with dynamic and structured attention patterns in dLLMs.
- We propose Focus-dLLM, a novel training-free acceleration framework that consists of a past confidence-guided indicator for predicting the next unmasked positions with a sink-aware dynamic token pruning strategy for efficient sparse attention.
- Experiments show that Focus-dLLM achieves substantial speedups over baselines while preserving accuracy. For instance, it attains better-than-vanilla performance and delivers $2.05\times$ speedup over Fast-dLLM for UltraLLaDA at $32K$ context length.

2 Related Work

Diffusion large language models. Diffusion large language models (dLLMs) (Li et al., 2025; You et al., 2025; Chen et al., 2025) have emerged as a promising non-autoregressive paradigm that enables parallel token generation via iterative denoising. Prior works explore both continuous-space diffusion for text (Li et al., 2022; Gong et al., 2022) and discrete-token diffusion formulations (Austin et al., 2021; Lou et al., 2023; He et al., 2023). Recent masked diffusion LMs (Nie et al., 2025; Zhu et al., 2025; Ye et al., 2025) have been successfully scaled up, demonstrating competitive performance against autoregressive counterparts at billion-parameter scales. Besides, long-context capability (Liu et al., 2025a; He et al., 2025) for dLLMs has also been explored, which pushes the context window up to $\geq 16K$ tokens.

KV cache for dLLMs. Due to bidirectional attention and token states evolving across denoising steps, dLLMs cannot directly reuse standard KV cache, motivating a line of caching-based accelerations (Ma et al., 2025; Huang et al., 2025e). Fast-dLLM (Wu et al., 2025) enables approximate KV reuse with block-wise strategies, while others (Ma et al., 2025; Liu et al., 2025b; Huang et al., 2025a) exploit dLLM-specific redundancy to reduce repeated computation (Huang et al., 2025f). More adaptive schemes (Jiang et al., 2025; Nguyen-Tri et al., 2025) further refine cache update granularity and timing. Nevertheless, accurately identifying which tokens require refresh in the next step remains challenging, and long-context inference still incurs substantial computation overhead under caching mechanisms.

Sparse attention for dLLMs. Attention sparsification (Zhang et al., 2025b,a,c), orthogonal to the KV cache mechanism, has also been explored to accelerate dLLM inference. Sparse-dLLM (Song et al., 2025) proposes dynamic cache eviction for diffusion decoding, but it adopts coarse and suboptimal block-level metrics. SparseD (Wang et al., 2025) reuses prior sparse patterns, yet it still relies on dense attention in early steps, restricting speedups. Moreover, these approaches do not account for the dynamic attention-sink behavior (Xiao et al., 2023) observed in dLLMs (Rulli et al., 2025). In contrast, our dynamic KV cache compression scheme adapts to step-varying contextual needs while preserving attention sinks for more efficient and accurate long-context inference.

3 Preliminaries

Diffusion LLM inference. Unlike autoregressive models that generate tokens sequentially, dLLMs generate text by iteratively denoising a fixed-length sequence. Let $\mathcal{V}$ denote the vocabulary and $\texttt{[MASK]} \in \mathcal{V}$ the special mask token. Given a prompt $\mathbf{p} = [p_1, \ldots, p_M]$, inference initializes at step 0 a length-L sequence by appending $N = L - M$ masks:

$$\mathbf{x}^{(T)} = [\underbrace{p_1, \ldots, p_M}_{\text{Prompt}}, \underbrace{\texttt{[MASK]}, \ldots, \texttt{[MASK]}}_{N=L-M}], \tag{1}$$

Let $\mathcal{M}^{(t)}$ denote the set of masked positions at denoising step t, where $\mathcal{M}^{(0)} = \{M+1, \ldots, L\}$ at initialization. The decoding process then iterates from $t = 0$ to $T - 1$. In step t, given the current sequence $\mathbf{x}^{(t)}$, the model f_θ produces a conditional distribution $p(x_i \mid \mathbf{x}^{(t)})$ for each masked position $i \in \mathcal{M}^{(t)}$. Then, a confidence-driven strategy (Nie et al., 2025; Ye et al., 2025) computes the predicted token $\hat{x}_i^{(t)}$ and its corresponding confidence score $c_i^{(t)}$ for each masked position i:

$$\begin{aligned} \hat{x}_i^{(t)} &= \arg\max_{v \in \mathcal{V}} p(x_i = v \mid \mathbf{x}^{(t)}), \\ c_i^{(t)} &= \max_{v \in \mathcal{V}} p(x_i = v \mid \mathbf{x}^{(t)}). \end{aligned} \tag{2}$$

Last, this strategy unmasks the highest-confidence positions while remasking the rest.

Approximate KV cache in dLLMs. Bidirectional attention makes the KV cache mechanism not applicable for dLLMs. To reduce computation costs, recent studies (Wu et al., 2025; Liu et al., 2025b; Ma et al., 2025) exploit *Approximate* KV cache, which updates KV states for a selected subset of tokens while reusing cached states for the rest. Formally, let $\mathcal{U}^{(t)}$ be the token indices refreshed at step t. The Key state $\mathbf{K}_i^{(t)}$ (and similarly $\mathbf{V}_i^{(t)}$) is

$$\mathbf{K}_i^{(t)} = \begin{cases} f_K(\mathbf{x}^{(t)})_i, & i \in \mathcal{U}^{(t)} \quad \text{(Compute)} \\ \widetilde{\mathbf{K}}_i, & i \notin \mathcal{U}^{(t)} \quad \text{(Reuse)} \end{cases}, \tag{3}$$

where $\widetilde{\mathbf{K}}_i$ denotes the cached state from the previous iteration. $f_K(\mathbf{x}^{(t)})_i$ is the current computed Key state, which is also used to update the cache.

4 Motivation

In this section, we investigate the token-confidence consistency and attention patterns tailored for dLLMs. Both of them inspire the core design of our Focus-dLLM.

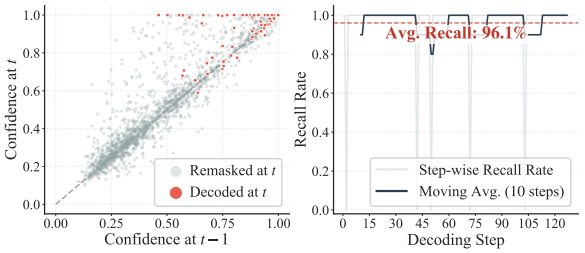

Figure 1: Confidence dynamics analysis for LLaDA-8B-Instruct (Nie et al., 2025) on GSM8K (Cobbe et al., 2021) ($L = 76$, $N = 128$, and $T = 128$). (*Left*) Confidence score correlation between adjacent steps. (*Right*) Step-wise recall rates of predicting the unmasked tokens at t using the remasked tokens with top-4 highest confidence scores at $t - 1$.

4.1 Temporal Consistency of Confidence

For dLLMs, effectively assessing the redundancy of attention computation *w.r.t.* tokens that are poised to be unmasked first requires locating these tokens in advance. To achieve this, we conduct a pivotal study related to their confidence score $c_i^{(t)}$. As illustrated in Figure 1 (*Left*), $c_i^{(t)}$ and $c_i^{(t-1)}$ correlate strongly in a positive manner. Also, the tokens that are to be decoded (unmasked) at t present a similarly high-confidence level in the preceding step $t - 1$. To quantitatively explore this relationship, we select the top-4 remasked tokens (*i.e.*, [MASK] at t) with the highest confidence scores at $t - 1$ and evaluate their overlap with the tokens decoded at the subsequent step t. As a result, Figure 1 (*Right*) shows a remarkably high average recall (96.1%) across decoding steps. These observations support the following key claim:

> The substantial overlap in confidence distributions reveals that tokens unmasked at t can be reliably located according to the confidence of tokens at the prior step $t - 1$.

4.2 Spatial Consistency of Attention Sinks

In this part, we explore the properties and variations of attention patterns for dLLMs. Similar to prior studies (Song et al., 2025; Rulli et al., 2025), as depicted in Figure 2, we also find that: (*i*) Attention maps exhibit strong locality, concentrating near the diagonal and favoring nearby context. (*ii*) Attention sinks (bright vertical bands), which strongly influence semantic continuity (Xiao et al., 2023; Gu et al., 2025), emerge and evolve across denoising steps. Due to the dynamics of these sinks, it is necessary to repeatedly identify their location to

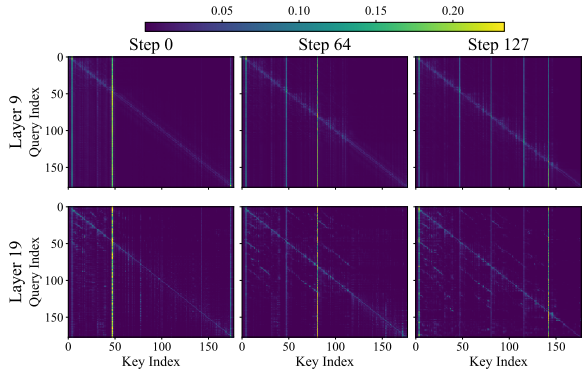

Figure 2: Attention patterns across decoding steps and layers in LLaDA-8B-Instruct (Nie et al., 2025) ($L = 49$, $N = 128$, $T = 128$). More visual results can be found in the Appendix.

preserve them in high-performing sparse attention. Despite this, we fortunately discovered a structured inter-layer consistency for attention sinks. To be specific, the index of attention sinks across different layers (*e.g.*, Layer 9 *vs*. Layer 19 in Figure 2) typically matches. Therefore, we believe that:

> Attention sinks with strong cross-layer consistency enable reliable identification at a certain intermediate depth, and the results can be reused for deeper layers to eliminate redundant computation.

5 Focus-dLLM

5.1 Framework Overview

In this section, we present the inference workflow of Focus-dLLM (Figure 3). Following Wu *et al.* (Wu et al., 2025), we adopt the KV caching mechanism with the semi-autoregressive remasking strategy (Nie et al., 2025) (*i.e.*, non-autoregressive unmasking within each block and autoregressive block-wise inference from left block to right block) for dLLMs.

For all blocks, Focus-dLLM performs a full cache refresh at each block entry step, which is commonly adopted in prior works (Wu et al., 2025; Song et al., 2025). For the other denoising steps, we use our proposed sparse attention pipeline to systematically prune redundancy:

- *Section 5.2:* Inspired by Section 4.1, we use confidence scores from the previous step to predict masked positions that are likely to be decoded, which provide focused queries for redundancy estimation in the latter Key/Value pruning. Additionally, guided by the locality pattern in Section 4.2, we expand these positions to local windows to form an active Query set and exclude the remaining Query tokens to compute the attention.

- *Section 5.3:* We accelerate inference via a sink-aware sparse attention mechanism. Since shallow layers are more sensitive to sparsification (Huang et al., 2025a), we treat the initial layers as *dense layers* with full attention. For subsequent *sparse layers*, we reuse the locations of attention sinks identified at the last dense layer. Finally, we apply dynamic block-wise pruning to Key/Value states of the prompt to keep the most relevant history, while retaining recognized sinks and all response tokens to preserve semantic coherence.

5.2 Past Confidence-Guided Indicator

Motivated by the temporal consistency analysis in Section 4.1, we introduce a *past confidence-guided indicator*, which adopts the confidence derived from step $t-1$ to accurately inform the tokens that are likely to be unmasked at step t. To be specific, among all positions $\mathcal{M}^{(t)}$ that remain in the `[MASK]` state within the current decoding block at t, we rank them by their prior confidence scores $c_j^{(t-1)}$ and select top-k indices as the candidate set $\mathcal{I}_{\text{focus}}$ to predict the future unmasked positions at t:

$$\mathcal{I}_{\text{focus}} = \left\{ i \mid c_i^{(t-1)} \in \text{top-}k\big(\{c_j^{(t-1)}\}_{j \in \mathcal{M}^{(t)}}\big) \right\}, \tag{4}$$

where $k = \lfloor \rho n^{(t)} \rceil$. $n^{(t)}$ is the number of tokens to be unmasked and ρ is a pre-defined prediction expansion factor. By leveraging the candidate set $\mathcal{I}_{\text{focus}}$, we can precisely determine the relevant history to prune redundant attention computation in the next subsection.

In addition, as discussed in Section 4.2, the attention mechanism in dLLMs exhibits a clear locality property, meaning that token representations depend strongly on nearby semantic context, while distant tokens typically contribute little. To leverage this property for computation savings, we propose a window expansion strategy that disregards the distant tokens and only preserves local windows for currently decoded Query tokens (positions in $\mathcal{I}_{\text{focus}}$) for attention computation. The position set

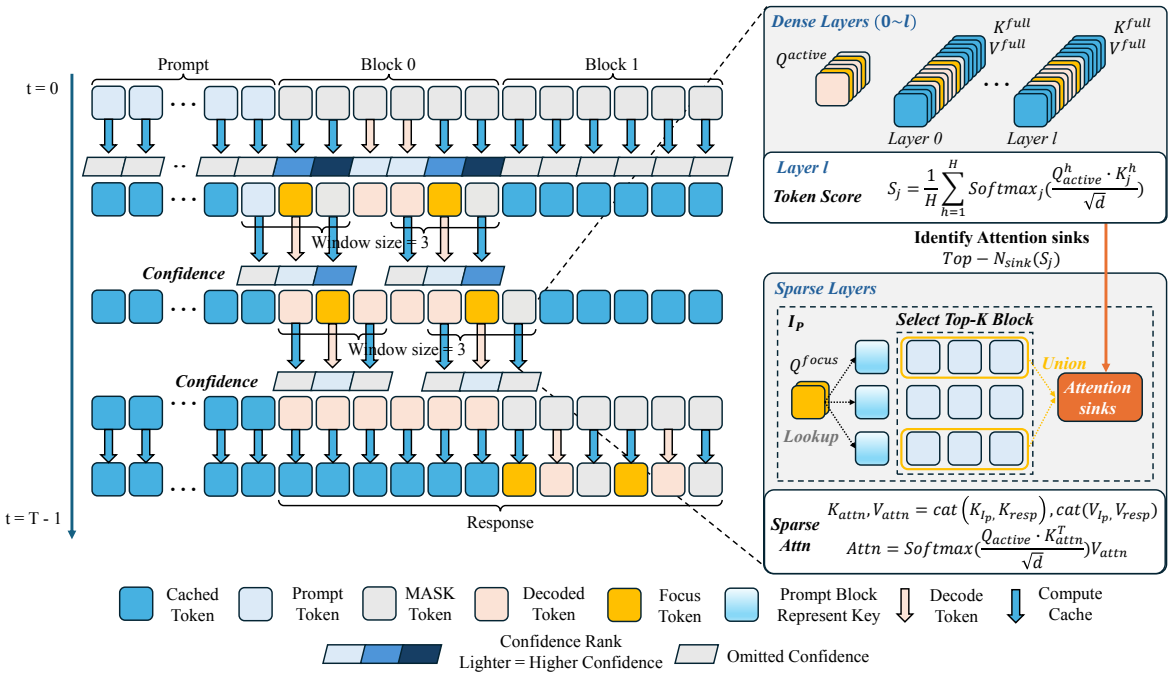

Figure 3: Overview of Focus-dLLM. We predict unmasked positions at the current step using previous confidence scores. These positions act as queries to retrieve relevant prompt blocks, where attention is computed over the union of these blocks and dynamically identified attention sinks.

corresponding to the union of windows is given as:

$$\mathcal{I}_{\text{active}} = \bigcup_{i \in \mathcal{I}_{\text{focus}}} \{l \mid i - \lfloor w/2 \rfloor \le l \le i + \lfloor w/2 \rfloor\}, \tag{5}$$

where w is the window size.

5.3 Sink-Aware Sparse Attention

Performing attention over the entire long-context history remains the primary computational bottleneck for inference. To address this, we propose a sink-aware sparse attention strategy that selectively retains only the most critical history for diffusion decoding.

Dynamic attention sinks identification. While retaining attention sinks is crucial for preserving generation quality (Xiao et al., 2024b; Rulli et al., 2025), existing sparse approaches for dLLMs (Song et al., 2025) typically overlook this, thereby risking the discard of tokens pivotal for generation quality (Rulli et al., 2025). Motivated by our observation in Section 4.2, we propose to explicitly identify and retain them. Crucially, this strategy shares the identified sink tokens across layers, avoiding redundant re-calculation at every depth.

Specifically, we designate the first l_{dense} layers as dense layers that perform full attention. Due to the cross-layer consistency observed in Section 4.2, we utilize the attention distribution at the cut-off layer l_{dense} as a reliable probe to identify globally salient tokens for the subsequent sparse layers. Let $\mathcal{I}_{\text{active}}$ denote the active token set obtained from Section 5.2. We define the aggregated query representation over active tokens as $Q_{\mathcal{I}_{\text{active}}}$. The importance score of each token j is computed as:

$$S_j = \frac{1}{H} \sum_{h=1}^{H} \text{Softmax}_j \left(\frac{Q^h_{\mathcal{I}_{\text{active}}} \cdot K^h_j}{\sqrt{d}} \right), \tag{6}$$

where H denotes the number of attention heads.

We then select the top-N_{sink} tokens with the highest scores to form the dynamic attention sink set, denoted as $\mathcal{I}_{\text{sink}} = \text{Top-}N_{\text{sink}}(S_j)$.

Block-wise token pruning. To accelerate inference while maximizing GPU efficiency, we implement block-wise token pruning to reduce computational overhead. Specifically, we partition the prompt tokens into contiguous blocks and assign each block a lightweight representative key, computed as the mean of the Key states within the block, $\bar{K}_b = \text{Mean}_{j \in \text{Block}_b}(K_j)$.

At timestep t, we estimate the relevance between the predicted candidate queries and each prompt block by aggregating their attention interactions. Concretely, for each block b, we compute a rele-

Table 1: Performance comparison on LongBench (Bai et al., 2024). **Bold** indicates the best performance among acceleration methods, and underlined indicates the second best.

Method	Single-Doc. QA		Multi-Doc. QA			Summarization		Few-shot Learning		Synthetic		Code		Ave. Score
	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	TREC	TriviaQA	Lsht	PRe	Lcc	RB-P	
UltraLLaDA (He et al., 2025)														
Vanilla	19.14	25.87	16.27	18.00	12.08	32.83	22.48	80.00	91.58	41.00	96.75	68.23	59.50	44.90
Fast-dLLM	18.34	**29.90**	17.03	17.11	13.36	30.05	22.89	**79.50**	91.03	**42.00**	94.75	67.50	58.10	44.74
Sparse-dLLM	18.04	27.26	20.59	17.88	13.67	29.95	**23.57**	76.50	**91.93**	41.50	**97.12**	67.50	57.72	44.86
SparseD	**19.09**	25.87	15.45	18.04	11.92	**32.64**	22.50	**79.50**	90.70	41.50	96.79	**68.10**	**59.02**	44.70
Focus-dLLM	17.02	29.11	**22.47**	**21.49**	**20.20**	26.75	21.45	77.00	90.78	41.00	95.73	66.72	57.14	**45.14**
Dream-7B-Instruct (Ye et al., 2025)														
Vanilla	35.58	40.49	41.59	42.10	23.36	23.51	19.66	71.50	87.34	15.75	32.50	63.79	62.23	43.03
Fast-dLLM	37.54	**43.24**	35.56	35.74	17.97	21.14	19.57	70.50	88.25	16.75	46.17	62.21	61.11	42.75
Sparse-dLLM	37.50	43.23	36.83	34.97	17.05	20.60	**20.05**	70.00	88.38	17.00	**46.50**	62.80	61.20	42.78
SparseD	**37.66**	40.96	**41.39**	**41.61**	**24.17**	**23.51**	19.66	**72.50**	86.85	15.75	36.83	**63.86**	**61.98**	**43.59**
Focus-dLLM	37.38	41.96	38.96	38.56	18.05	21.06	19.26	70.00	**88.76**	**17.25**	44.25	60.62	60.50	42.82

vance score as

$$R_b = \frac{1}{H} \sum_{h=1}^{H} \left(Q^h_{\mathcal{I}_{\text{focus}}} \cdot \bar{K}^h_b \right), \tag{7}$$

where $\mathcal{I}_{\text{focus}}$ denotes the predicted candidate set obtained in Section 5.2.

Based on these relevance scores, we select the top $C = \lfloor \alpha \cdot N_{\text{total\_blocks}} \rfloor$ blocks to form the set of relevant blocks, $\mathcal{B}_{\text{relevant}} = \text{Top-}C(R_b)$. The final attention index set is constructed as the union of dynamically identified attention sinks and tokens within the selected relevant prompt blocks:

$$\mathcal{I}_p = \mathcal{I}_{\text{sink}} \cup \bigcup_{b \in \mathcal{B}_{\text{relevant}}} \{\, i \mid i \in \text{Block}_b \,\}. \tag{8}$$

Using this index set, we perform sparse attention by gathering keys and values exclusively from the selected prompt tokens and the response tokens. Specifically, for the active queries $\mathcal{Q}_{\mathcal{I}_{\text{active}}}$, the effective Key-Value pairs are formed as: $K_{\text{attn}} = \text{concat}(K_{\mathcal{I}_p}, K_{\text{resp}})$, $V_{\text{attn}} = \text{concat}(V_{\mathcal{I}_p}, V_{\text{resp}})$. The resulting sparse attention is then computed as:

$$\text{Attn} = \text{Softmax}\left(\frac{Q_{\mathcal{I}_{\text{active}}} K^\top_{\text{attn}}}{\sqrt{d}} \right) V_{\text{attn}}. \tag{9}$$

6 Experiments

6.1 Experiments Settings

Models. We evaluate our method on two representative diffusion LLMs: UltraLLaDA (He et al., 2025) and Dream-7B-Instruct (Ye et al., 2025).

Baselines. We compare Focus-dLLM against standard native inference (Vanilla) and three dLLM acceleration frameworks: Fast-dLLM (Wu et al., 2025), SparseD (Wang et al., 2025), and Sparse-dLLM (Song et al., 2025).

Benchmarks. To comprehensively assess long-context capabilities, we conduct evaluations on LongBench (Bai et al., 2024), a widely adopted benchmark specifically designed for multi-task long-context understanding.

Implementation details. All experiments were conducted on NVIDIA H200 GPUs using OpenCompass (Contributors, 2023). To ensure a fair comparison, all baselines utilize the recommended configurations provided in their official implementations. Specifically, for SparseD, we set $skip = 20\%$, $ratio = 30\%$, and $block\_size = 128$; for Sparse-dLLM, we use retention ratio $r = 0.5$ and kernel size $s = 3$. For the Focus-dLLM setup, we adopt identical hyperparameters for both UltraLLaDA and Dream: we set prediction expansion factor $\rho = 4$, window size $w = 8$, dense layers $l_{\text{dense}} = 6$, and sparsity ratio $\alpha = 0.5$. Additionally, the number of sink tokens is set to $N_{\text{sinks}} = 0.01 \times M$, where M denotes the prompt length, and prompt block size = 64. Additional details for datasets, models, and methods are provided in the Appendix A.

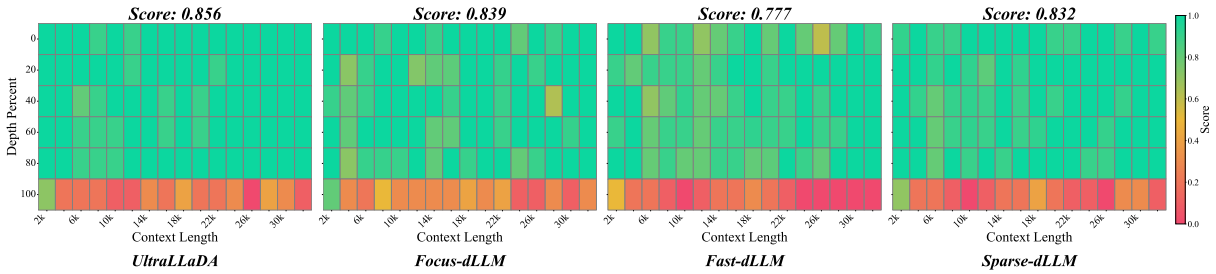

Figure 4: Niah (Kamradt, 2023) results on UltraLLaDA (He et al., 2025) under long-context settings with a maximum context length of $32K$ across different layer depths.

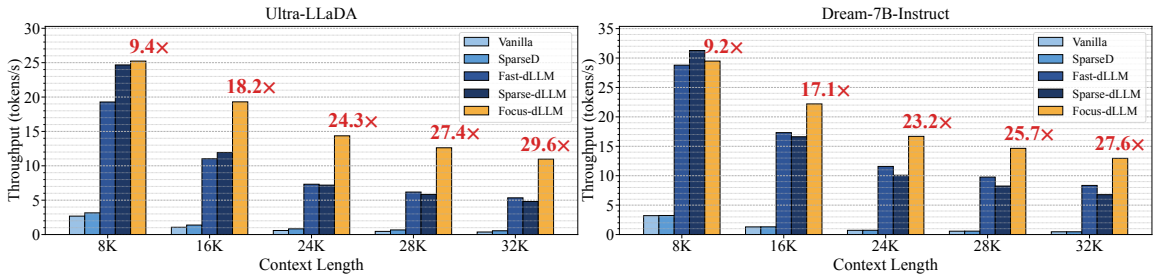

Figure 5: Efficiency evaluation. Comparison of decoding throughput (tokens/s) on UltraLLaDA (He et al., 2025) (*Left*) and Dream-7B-Instruct (Ye et al., 2025) (*Right*) across varying context lengths. Red numbers indicate the speedup ratio of Focus-dLLM relative to the Vanilla baseline.

6.2 Main Results

Accuracy. As presented in Table 1, Focus-dLLM demonstrates robust performance across both evaluated diffusion models. On UltraLLaDA (He et al., 2025), our method achieves the highest average score, outperforming the Vanilla baseline and all competing acceleration frameworks. On Dream-7B-Instruct (Ye et al., 2025), Focus-dLLM again surpasses Spare-dLLM (Song et al., 2025) and Fast-dLLM (Wu et al., 2025), performing on par with the Vanilla baseline. While its accuracy is marginally lower than SparseD (Wang et al., 2025), Focus-dLLM offers a compelling advantage in efficiency, achieving up to a $19.95\times$ speedup at a $32K$ (with 1K denoting 1024 tokens) context length (as shown in Figure 5). This highlights our method's superior balance between performance and inference speed, establishing it as a more practical solution.

Niah experiments. Figure 4 reports Niah (Kamradt, 2023) results on UltraLLaDA (He et al., 2025) under long-context settings with a maximum context length of $32K$. Focus-dLLM achieves overall higher scores than Fast-dLLM (Wu et al., 2025) and Sparse-dLLM (Song et al., 2025) across layers, and attains better accuracy than the vanilla baseline at the deepest layer, demonstrating strong needle-in-a-haystack retrieval.

Efficiency. We evaluate the scalability of Focus-dLLM by measuring throughput across context lengths from $8K$ to $32K$ context length, both the generation length and generation steps are fixed at 256. As shown in Figure 5, our method consistently outperforms all baselines, with the speedup ratio over Vanilla notably expanding as context grows—from $9.4\times$ at $8K$ context length to $29.6\times$ at $32K$ context length. This trend can be attributed to the reduction of redundant attention computation, which tends to incur more significant overhead as sequences lengthen. Consequently, Focus-dLLM maintains superior efficiency and surpasses existing frameworks like Fast-dLLM by up to $2.05\times$ at $32K$ context length.

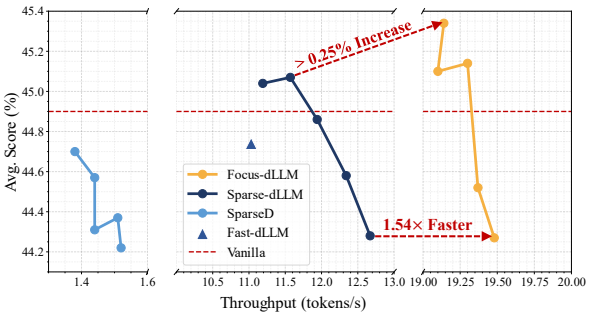

Figure 6: Accuracy *vs.* throughput for UltraLLaDA (He et al., 2025) on LongBench (Bai et al., 2024) with $16K$.

Accuracy *vs.* efficiency. Figure 6 compares decoding throughput and LongBench (Bai et al., 2024)

accuracy across different methods and configurations, with throughput measured at a $16K$ context length. Focus-dLLM consistently forms a stronger Pareto frontier than prior approaches, achieving higher throughput with comparable or better accuracy. Additional experimental details and configurations are provided in the Appendix B.

7 Ablation Study

Table 2: Ablation study of Focus-dLLM on UltraLLaDA (He et al., 2025). PCGI denotes Past Confidence-Guided Indicator, and SA Sparse Attn represents sink-aware sparse attention.

Method	Avg. Score	Throughput
Fast-dLLM	44.74	11.03
+ PCGI	44.23 -0.51	11.37 +0.34
+ SA Sparse Attn	44.84 +0.10	17.68 +6.65
Focus-dLLM	**45.14** +0.40	**17.71** +6.68

Effectiveness of each component. We evaluate the impact of the proposed components on LongBench average score and $16K$ context decoding throughput. Table 2 presents the ablation results building on the Fast-dLLM (Wu et al., 2025) baseline. PCGI filters active queries via our past confidence-guided indicator while attending to the full context KV. SA Sparse Attn prunes context, while passing the entire block as active tokens and identifies redundancy for pruning using these tokens. Applying PCGI alone slightly degrades accuracy, while SA Sparse Attn improves accuracy by filtering irrelevant tokens in long contexts and significantly increasing throughput. Combining both components, Focus-dLLM achieves further accuracy gains and the highest throughput, demonstrating that accurate query selection enables more precise and effective sparse attention.

Table 3: Effect of attention sinks on LongBench accuracy for Dream-7B-Instruct (Ye et al., 2025). Incorporating attention sinks consistently improves performance across tasks.

Subset	w/o Attn Sinks	w/ Attn Sinks
hotpotqa	37.17	38.96+1.79
2wikimqa	37.68	38.56+0.88
trec	69.50	70.00+0.50
Avg. Score	41.47	**42.82**+1.35

Table 3 evaluates the effectiveness of attention sinks on Dream-7B-Instruct (Ye et al., 2025). Incorporating attention sinks leads to a clear improvement on LongBench (Bai et al., 2024). These results suggest that effectively retaining attention sinks contributes to the preservation of key contextual information.

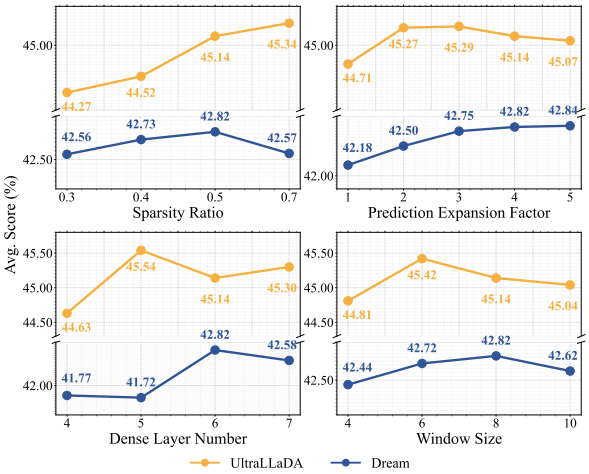

Figure 7: Ablations on hyperparameters of Focus-dLLM on LongBench (Bai et al., 2024).

Ablations on hyperparameters. Figure 7 analyzes the impact of key hyperparameters in Focus-dLLM. Increasing the sparsity ratio α generally improves accuracy, indicating that retaining more relevant context benefits long-context reasoning, while the drop observed for Dream (Ye et al., 2025) at α=0.7 suggests that excessive retention may introduce irrelevant context and dilute useful signals. For the prediction expansion factor ρ, small values (*e.g.*, 1) lead to poor accuracy due to insufficient recall of future decoded token positions, whereas larger values provide more reliable coverage and steadily improve performance. Varying the number of dense layers l_{dense} results in non-monotonic behavior, implying that attention sinks are not fully stabilized in shallow layers and that sparsification sensitivity differs across depths. A similar trend is observed for the window size w: overly small windows miss necessary local context, moderate windows improve accuracy, and excessively large windows degrade performance by introducing unrelated tokens.

8 Conclusion

We analyzed diffusion inference dynamics and introduced Focus-dLLM, a training-free framework for accelerating long-context dLLM inference. By leveraging a *past confidence-guided indicator* for query prediction and a *sink-aware pruning strategy* to retain critical history, our method effectively

eliminates redundant computation. Experiments demonstrate that Focus-dLLM achieves over $29\times$ speedup at $32K$ context length while maintaining superior performance compared to state-of-the-art baselines.

Limitations

While Focus-dLLM demonstrates high efficiency in text tasks, its extension to multimodal reasoning remains a direction for future exploration. Additionally, our current hyperparameters are manually configured, which may not achieve optimal performance across all specialized domains. Developing a fully adaptive mechanism for dynamic parameter adjustment represents a promising avenue to further enhance the framework's versatility and robustness.

Acknowledgments

This work is supported in part by the Beijing Natural Science Foundation (Grant No. L243031), the National Key R&D Program of China (Grant No. 2024YFB4505601), and the National Natural Science Foundation of China (Grant No. 62572036).

References

Marianne Arriola, Aaron Gokaslan, Justin T. Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2025. Block diffusion: Interpolating between autoregressive and diffusion language models. *Preprint*, arXiv:2503.09573.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, and 1 others. 2024. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 3119–3137.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, Chengxi Li, Chongxuan Li, Jianguo Li, Zehuan Li, Huabin Liu, Lin Liu, Guoshan Lu, Xiaocheng Lu, Yuxin Ma, and 12 others. 2025. Llada2.0: Scaling up diffusion language models to 100b. *Preprint*, arXiv:2512.15745.

Sitong Chen, Shen Nie, Jiacheng Sun, Zijin Feng, Zhenguo Li, Ji-Rong Wen, and Chongxuan Li. 2025. Masked diffusion models as energy minimization. *arXiv preprint arXiv:2509.13866*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Xingtong Ge, Yi Zhang, Yushi Huang, Dailan He, Xiahong Wang, Bingqi Ma, Guanglu Song, Yu Liu, and Jun Zhang. 2026. Salt: Self-consistent distribution matching with cache-aware training for fast video generation. *arXiv preprint arXiv:2604.03118*.

Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Chengtao Lv, Yunchen Zhang, Dacheng Tao, and Xianglong Liu. 2024. Llmc: Benchmarking large language model quantization with a versatile compression toolkit. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 132–152.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. 2025. Scaling diffusion language models via adaptation from autoregressive models. *Preprint*, arXiv:2410.17891.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Xiangming Gu, Tianyu Pang, Chao Du, Qian Liu, Fengzhuo Zhang, Cunxiao Du, Ye Wang, and Min Lin. 2025. When attention sink emerges in language models: An empirical view. *Preprint*, arXiv:2410.10781.

Guangxin He, Shen Nie, Fengqi Zhu, Yuankang Zhao, Tianyi Bai, Ran Yan, Jie Fu, Chongxuan Li, and Binhang Yuan. 2025. Ultrallada: Scaling the context length to 128k for diffusion large language models. *Preprint*, arXiv:2510.10481.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. 2023. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the*

61st annual meeting of the association for computational linguistics (volume 1: Long papers), pages 4521–4534.

Jianuo Huang, Yaojie Zhang, Yicun Yang, Benhao Huang, Biqing Qi, Dongrui Liu, and Linfeng Zhang. 2025a. Mask tokens as prophet: Fine-grained cache eviction for efficient dllm inference. *Preprint*, arXiv:2510.09309.

Yushi Huang, Xingtong Ge, Ruihao Gong, Chengtao Lv, and Jun Zhang. 2025b. Linvideo: A post-training framework towards o (n) attention in efficient video generation. *arXiv preprint arXiv:2510.08318*.

Yushi Huang, Ruihao Gong, Jing Liu, Tianlong Chen, and Xianglong Liu. 2024. Tfmq-dm: Temporal feature maintenance quantization for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7362–7371.

Yushi Huang, Ruihao Gong, Jing Liu, Yifu Ding, Chengtao Lv, Haotong Qin, and Jun Zhang. 2025c. Qvgen: Pushing the limit of quantized video generative models. *arXiv preprint arXiv:2505.11497*.

Yushi Huang, Ruihao Gong, Xianglong Liu, Jing Liu, Yuhang Li, Jiwen Lu, and Dacheng Tao. 2025d. Temporal feature matters: A framework for diffusion model quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yushi Huang, Zining Wang, Ruihao Gong, Jing Liu, Xinjie Zhang, Jinyang Guo, Xianglong Liu, and Jun Zhang. 2025e. Harmonica: Harmonizing training and inference for better feature caching in diffusion transformer acceleration. In *International Conference on Machine Learning*, pages 25835–25858. PMLR.

Yushi Huang, Zining Wang, Zhihang Yuan, Yifu Ding, Ruihao Gong, Jinyang Guo, Xianglong Liu, and Jun Zhang. 2025f. Modes: Accelerating mixture-of-experts multimodal large language models via dynamic expert skipping. *arXiv preprint arXiv:2511.15690*.

Yuchu Jiang, Yue Cai, Xiangzhong Luo, Jiale Fu, Jiarui Wang, Chonghan Liu, and Xu Yang. 2025. d^2cache: Accelerating diffusion-based llms via dual adaptive caching. *Preprint*, arXiv:2509.23094.

G. Kamradt. 2023. Needle in a haystack - pressure testing llms. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. 2025. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Xiaoran Liu, Yuerong Song, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025a. Longllada: Unlocking long context capabilities in diffusion llms. *Preprint*, arXiv:2506.14429.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. 2025b. dllm-cache: Accelerating diffusion large language models with adaptive caching. *Preprint*, arXiv:2506.06295.

Lingkun Long, Rubing Yang, Yushi Huang, Desheng Hui, Ao Zhou, and Jianlei Yang. 2026. Sliminfer: Accelerating long-context llm inference via dynamic token pruning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32284–32292.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2023. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.

Chengtao Lv, Yumeng Shi, Yushi Huang, Ruihao Gong, Shen Ren, and Wenya Wang. 2026a. Light forcing: Accelerating autoregressive video diffusion via sparse attention. *arXiv preprint arXiv:2602.04789*.

Chengtao Lv, Bilang Zhang, Yang Yong, Ruihao Gong, Yushi Huang, Shiqiao Gu, Jiajun Wu, Yumeng Shi, Jinyang Guo, and Wenya Wang. 2026b. Llmc+: Benchmarking vision-language model compression with a plug-and-play toolkit. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 24189–24197.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. 2025. dkv-cache: The cache for diffusion language models. *Preprint*, arXiv:2505.15781.

Quan Nguyen-Tri, Mukul Ranjan, and Zhiqiang Shen. 2025. Attention is all you need for kv cache in diffusion llms. *Preprint*, arXiv:2510.14973.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large language diffusion models. *Preprint*, arXiv:2502.09992.

Maximo Eduardo Rulli, Simone Petruzzi, Edoardo Michielon, Fabrizio Silvestri, Simone Scardapane, and Alessio Devoto. 2025. Attention sinks in diffusion language models. *Preprint*, arXiv:2510.15731.

Valeria Ruscio, Umberto Nanni, and Fabrizio Silvestri. 2025. What are you sinking? a geometric approach on attention sink. *Preprint*, arXiv:2508.02546.

Jay Shah, Ganesh Bikshandi, Ying Zhang, Vijay Thakkar, Pradeep Ramani, and Tri Dao. 2024. Flashattention-3: Fast and accurate attention with asynchrony and low-precision. *Advances in Neural Information Processing Systems*, 37:68658–68685.

Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *Preprint*, arXiv:2508.02558.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024. Quest: Query-aware sparsity for efficient long-context llm inference. *arXiv preprint arXiv:2406.10774*.

Philippe Tillet, Hsiang-Tsung Kung, and David Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. In *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*, pages 10–19.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Zeqing Wang, Gongfan Fang, Xinyin Ma, Xingyi Yang, and Xinchao Wang. 2025. Sparsed: Sparse attention for diffusion language models. *arXiv preprint arXiv:2509.24014*.

Zining Wang, Jinyang Guo, Ruihao Gong, Yang Yong, Aishan Liu, Yushi Huang, Jiaheng Liu, and Xianglong Liu. 2024. Ptsbench: A comprehensive post-training sparsity benchmark towards algorithms and models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5742–5751.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *Preprint*, arXiv:2505.22618.

Chaojun Xiao, Pengle Zhang, Xu Han, Guangxuan Xiao, Yankai Lin, Zhengyan Zhang, Zhiyuan Liu, and Maosong Sun. 2024a. Infllm: Training-free long-context extrapolation for llms with an efficient context memory. *Advances in Neural Information Processing Systems*, 37:119638–119661.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Ruyi Xu, Guangxuan Xiao, Haofeng Huang, Junxian Guo, and Song Han. 2025. XAttention: Block sparse attention with antidiagonal scoring. In *Forty-second International Conference on Machine Learning*.

Yufei Xue, Yushi Huang, Jiawei Shao, and Jun Zhang. 2025. Vlmq: Efficient post-training quantization for large vision-language models via hessian augmentation. *arXiv e-prints*, pages arXiv–2508.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7b: Diffusion large language models. *Preprint*, arXiv:2508.15487.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. 2025. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*.

Jingyang Yuan, Huazuo Gao, Damai Dai, Junyu Luo, Liang Zhao, Zhengyan Zhang, Zhenda Xie, Y. X. Wei, Lean Wang, Zhiping Xiao, Yuqing Wang, Chong Ruan, Ming Zhang, Wenfeng Liang, and Wangding Zeng. 2025. Native sparse attention: Hardware-aligned and natively trainable sparse attention. *Preprint*, arXiv:2502.11089.

Jintao Zhang, Haofeng Huang, Pengle Zhang, Jia Wei, Jun Zhu, and Jianfei Chen. 2025a. Sageattention2: Efficient attention with thorough outlier smoothing and per-thread int4 quantization. In *International Conference on Machine Learning (ICML)*.

Jintao Zhang, Jia Wei, Pengle Zhang, Jun Zhu, and Jianfei Chen. 2025b. Sageattention: Accurate 8-bit attention for plug-and-play inference acceleration. In *International Conference on Learning Representations (ICLR)*.

Jintao Zhang, Chendong Xiang, Haofeng Huang, Jia Wei, Haocheng Xi, Jun Zhu, and Jianfei Chen. 2025c. Spargeattn: Accurate sparse attention accelerating any model inference. In *International Conference on Machine Learning (ICML)*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *Preprint*, arXiv:2505.19223.

Lunjie Zhu, Yushi Huang, Xingtong Ge, Yufei Xue, Zhening Liu, Yumeng Zhang, Zehong Lin, and Jun Zhang. 2026. Flash-vaed: Plug-and-play vae decoders for efficient video generation. *arXiv preprint arXiv:2602.19161*.

Appendix

A Implementation Details

A.1 Details of Focus-dLLM

This section provides additional details regarding the implementation of our Focus-dLLM framework. To maximize computational efficiency (Wang et al., 2024; Lv et al., 2026b; Gong et al., 2024; Huang et al., 2025d), our framework leverages specialized GPU kernels for attention computations. The core sink-aware sparse attention operator (Lv et al., 2026a; Ge et al., 2026), which handles dynamic context pruning, is implemented using Triton (Tillet et al., 2019). This allows for fine-grained control (Xue et al., 2025) and optimization of memory access patterns for sparse matrix operations. For dense attention computations—which occur in the initial dense layers ($l \leq l_{\text{dense}}$) and during full-cache refreshes at block entries—we utilize the highly optimized FlashAttention (Shah et al., 2024) kernel to accelerate inference.

Empirical analysis revealed that the final layers of Dream-7B-Instruct (Ye et al., 2025) exhibit high sensitivity to attention sparsification. To mitigate potential performance degradation, we designate the final four transformer layers of this model as dense, ensuring they always perform full attention (Long et al., 2026). This hybrid strategy preserves the integrity of critical generation stages, achieving a superior trade-off between accuracy and efficiency.

Focus-dLLM strictly adhere to the original decoding strategies of both UltraLLaDA (He et al., 2025) and Dream-7B-Instruct (Ye et al., 2025). The generation process follows the semi-autoregressive remasking paradigm, where a transfer scheduler dictates which tokens are unmasked at each step based on their confidence scores, consistent with the methods described in (He et al., 2025) and (Ye et al., 2025). The complete inference procedure of Focus-dLLM is detailed in Algorithm 1.

A.2 Baselines

In our experiments, we compare Focus-dLLM against the vanilla inference of representative diffusion models and several state-of-the-art acceleration frameworks.

Vanilla dLLMs. We use the standard inference implementations of UltraLLaDA (He et al., 2025) and Dream-7B-Instruct (Ye et al., 2025) as our primary baselines. UltraLLaDA is developed by fine-tuning LLaDA (Nie et al., 2025) for long-context capabilities, while Dream is adapted from a pre-trained autoregressive model. These representative diffusion LLMs perform a full attention computation over the entire sequence at each denoising step, without any caching or sparsification mechanisms (Huang et al., 2025b).

Fast-dLLM. As a strong baseline for approximate KV cache methods, Fast-dLLM (Wu et al., 2025) introduces a block-wise approximate KV cache tailored for the bidirectional attention in dLLMs. It reuses cached activations from previously decoded blocks to reduce redundant computation.

Sparse-dLLM. This method (Song et al., 2025) accelerates dLLM inference by integrating dynamic cache eviction with sparse attention principles. It leverages the temporal stability of token saliency to identify and retain critical KV entries while dynamically evicting unimportant entries from both the prefix and suffix contexts.

SparseD. As a pure sparse attention baseline, SparseD (Wang et al., 2025) is tailored for the unique attention patterns in dLLMs. Its core strategy involves pre-computing head-specific sparse patterns once and reusing them across subsequent denoising steps. To preserve generation quality, it applies full attention during the critical early steps before switching to the pre-computed sparse patterns for the remainder of the inference process.

To ensure a fair comparison, all methods uniformly employ the semi-autoregressive remasking strategy, with the block length set to 32 across all experiments.

A.3 Generation Settings

Table 4 presents the detailed configurations for each task in the LongBench benchmark. To align with the evaluation settings of UltraLLaDA (He et al., 2025), we process all input contexts by truncating them to a maximum length of $16K$ tokens using the "drop-middle" strategy. For each specific task, the generation length (Gen. Len.) and the generation steps (Steps) are configured as specified in the table.

B Details of Accuracy *vs.* Efficiency Experiments

This section provides the detailed results underpinning our accuracy vs. efficiency analysis. Table 5 presents a comprehensive performance comparison on the LongBench (Bai et al., 2024) for

Algorithm 1 Focus-dLLM Inference Procedure

Require: Prompt $\mathbf{p}$, Mask token [MASK], Max steps T, Sparse ratio α, Expansion factor ρ, Transfer Scheduler $\mathcal{S}$, Dense layers l_{dense}, Window size w.
Ensure: Generated sequence $\mathbf{x}^{(T)}$

Initialize $\mathbf{x}^{(0)} \leftarrow [\mathbf{p}, [\text{MASK}]_1, \ldots, [\text{MASK}]_N]$
Initialize $\mathbf{K}, \mathbf{V}$ cache as empty; Confidence scores $\mathbf{c}^{(0)} \leftarrow \mathbf{0}$
for $t = 0$ to $T - 1$ **do**
  ▷ Determine dynamic prediction token counts
  $n^{(t)} \leftarrow$ Number of tokens to unmask at step t given $\mathcal{S}$
  $k \leftarrow \lfloor \rho \cdot n^{(t)} \rceil$ ▷ Calculate candidate count
  if IsBlockEntry(t) **then**
    $\mathcal{I}_{\text{active}} \leftarrow \{1, \ldots, L\}$ ▷ Full refresh at block entry
    $use\_sparse \leftarrow$ False
  else
    $\mathcal{I}_{\text{focus}} \leftarrow$ Select top-k indices based on $\mathbf{c}^{(t)}$ ▷ Candidate set
    $\mathcal{I}_{\text{active}} \leftarrow \bigcup_{i \in \mathcal{I}_{\text{focus}}} \{i - \lfloor w/2 \rfloor, \ldots, i + \lfloor w/2 \rfloor\}$ ▷ Window expansion
    $use\_sparse \leftarrow$ True
  end if
  ▷ Layer-wise Forward Pass
  for layer $l = 1$ to L_{layers} **do**
    if use_sparse **and** $l > l_{\text{dense}}$ **then**
      ▷ Sparse Attention Mechanism
      $\mathcal{I}_{\text{sink}} \leftarrow$ IdentifySinks(Layer l_{dense})
      Compute Block Relevance R_b using $\mathbf{Q}_{\mathcal{I}_{\text{focus}}}$ and $\bar{\mathbf{K}}_b$
      Determine selection size $C = \lfloor \alpha \cdot N_{\text{total\_blocks}} \rfloor$
      Select relevant prompt blocks $\mathcal{B}_{\text{relevant}} \leftarrow$ Top-C(R_b)
      $\mathcal{I}_p \leftarrow \mathcal{I}_{\text{sink}} \cup \bigcup_{b \in \mathcal{B}_{\text{relevant}}} \{\, i \mid i \in \text{Block}_b \,\}$
      $\mathbf{K}_{\text{attn}} \leftarrow \text{Concat}(\mathbf{K}_{\mathcal{I}_p}, \mathbf{K}_{\text{resp}})$
      $\mathbf{V}_{\text{attn}} \leftarrow \text{Concat}(\mathbf{V}_{\mathcal{I}_p}, \mathbf{V}_{\text{resp}})$
      $\mathbf{H}_l \leftarrow \text{Softmax}\left(\frac{\mathbf{Q}_{\mathcal{I}_{\text{active}}} \mathbf{K}_{\text{attn}}^\top}{\sqrt{d}}\right) \mathbf{V}_{\text{attn}}$
    else
      ▷ Full Attention & Cache Update
      $\mathbf{H}_l \leftarrow \text{FullAttn}(\mathbf{Q}_{\mathcal{I}_{\text{active}}}, \mathbf{K}, \mathbf{V})$
      Update KV Cache for indices in $\mathcal{I}_{\text{active}}$
    end if
  end for
  ▷ Denoising and State Update
  Update $\mathbf{x}^{(t)}$ to $\mathbf{x}^{(t+1)}$ and compute new confidence $\mathbf{c}^{(t+1)}$
end for
return $\mathbf{x}^{(T)}$

UltraLLaDA (He et al., 2025), including various configurations for both baseline methods and our own. For Sparse-dLLM (Song et al., 2025), we vary the retention ratio r, which determines the percentage of KV cache entries preserved. For SparseD (Wang et al., 2025), configurations differ in the skip ratio (the initial portion of steps using full attention) and the selection ratio r. The configurations for our method, Focus-dLLM, correspond to different settings of the sparsity ratio α , which controls the amount of prompt context retained for attention computation, while all other hyperparameters remain consistent with the setup described in the main text(section 6.1). As the results consistently demonstrate, Focus-dLLM establishes a better accuracy-efficiency trade-off, achieving superior overall performance compared to prior acceleration methods.

Table 4: Detailed information of the datasets in the LongBench benchmark.

Label	Eval. Metric	Avg. Len.	Gen. Len.	Steps	Language	Sample Num.
Qasper	F1	3,619	32	32	EN	200
MultiFieldQA-en	F1	4,559	64	64	EN	150
HotpotQA	F1	9,151	32	32	EN	200
2WikiMQA	F1	4,887	32	32	EN	200
Musique	F1	11,214	32	32	EN	200
GovReport	Rouge-L	8,734	512	512	EN	200
QMSum	Rouge-L	10,614	512	512	EN	200
MultiNews	Rouge-L	2,113	512	512	EN	200
TREC	Accuracy	5,177	64	64	EN	200
TriviaQA	F1	8,209	32	32	EN	200
SAMSum	Rouge-L	6,258	128	128	EN	200
Lsht	Accuracy	22,333	64	64	ZN	200
PassageRetrieval	Accuracy	9,289	32	32	EN	200
Lcc	Edit Sim	1,235	64	64	Python/C#/Java	500
RepoBench-P	Edit Sim	4,206	64	64	Python/Java	500

C Attention Patterns of dLLM

To supplement our analysis in Section 4.2, Figure 8 presents a broader visualization of attention patterns from LLaDA-8B-Instruct (Nie et al., 2025) across various layers and denoising steps. The heatmaps clearly illustrate the principles of locality (strong diagonals) and the formation of attention sinks (bright vertical bands). Most importantly, the figure provides strong visual evidence for the cross-layer consistency of these sinks. The locations of the prominent vertical bands are remarkably stable across different layers (*e.g.*, Layer 9, 19, and 31) at any given denoising step. This observed stability is the primary motivation behind our method, as it validates our strategy of identifying sink locations at an intermediate depth and reusing them for deeper layers to eliminate redundant computation.

Table 5: Detailed performance and throughput comparison on LongBench (Bai et al., 2024) for UltraLLaDA (He et al., 2025). We report results for baselines and various configurations of our method, Focus-dLLM.

Method	Single-Doc. QA		Multi-Doc. QA			Summarization		Few-shot Learning		Synthetic		Code		Ave. Score	Throughput(16K)
	Qasper	MF-en	HotpotQA	2WikiMQA	Musique	GovReport	QMSum	TREC	TriviaQA	LSHT	PRe	Lcc	RB-P		
Vanilla	19.14	25.87	16.27	18.00	12.08	32.83	22.48	80.00	91.58	41.00	96.75	68.23	59.50	44.90	1.06
Fast-dLLM	18.34	29.90	17.03	17.11	13.36	30.05	22.89	79.50	91.03	42.00	94.75	67.50	58.10	44.74	11.03
Sparse-dLLM (r=0.3)	17.10	25.82	20.82	18.63	15.35	27.95	22.52	71.00	91.93	41.50	98.71	67.07	57.18	44.28	12.67
Sparse-dLLM (r=0.4)	17.99	27.67	18.90	18.55	13.11	28.77	23.01	74.50	91.43	42.00	98.17	67.80	57.68	44.58	12.34
Sparse-dLLM (r=0.5)	18.04	27.26	20.59	17.88	13.67	29.95	23.57	76.50	91.93	41.50	97.12	67.50	57.72	44.86	11.94
Sparse-dLLM (r=0.6)	19.06	26.94	20.80	18.30	14.31	30.16	23.68	77.00	91.43	42.50	96.25	67.99	57.44	45.07	11.57
Sparse-dLLM (r=0.7)	19.03	27.35	21.64	18.04	13.24	30.63	23.38	78.50	91.43	41.50	95.42	67.94	57.40	45.04	11.19
SparseD (skip=0.2,r=0.3)	19.09	25.87	15.45	18.04	11.92	32.64	22.50	79.50	90.70	41.50	96.79	68.10	59.02	44.70	1.38
SparseD (skip=0.2,r=0.2)	18.85	25.70	16.10	17.14	11.64	32.29	22.52	79.50	90.70	41.50	96.67	68.02	58.77	44.57	1.44
SparseD (skip=0.2,r=0.1)	18.06	25.76	15.40	16.64	11.47	31.44	22.60	79.50	91.05	41.50	96.84	67.98	58.59	44.37	1.51
SparseD (skip=0.1,r=0.3)	18.89	24.72	14.45	14.65	10.93	32.55	22.93	79.50	91.70	41.50	97.12	67.86	59.18	44.31	1.44
SparseD (skip=0.1,r=0.2)	19.34	24.21	14.09	13.84	10.80	31.61	23.12	79.50	91.53	42.00	97.88	67.72	59.16	44.22	1.52
Focus-dLLM (α=0.3)	16.63	29.57	23.09	21.14	18.59	26.09	21.05	69.50	91.28	38.00	97.27	66.19	57.07	44.27	19.48
Focus-dLLM (α=0.4)	16.60	27.83	23.81	21.34	18.52	26.85	21.03	72.00	90.78	40.00	97.23	66.55	56.18	44.52	19.37
Focus-dLLM (α=0.5)	17.02	29.11	22.47	21.49	20.20	26.75	21.45	77.00	90.78	41.00	95.73	66.72	57.12	45.14	19.30
Focus-dLLM (α=0.6)	16.91	28.20	22.75	23.21	18.94	26.39	21.86	76.50	90.78	41.50	95.84	66.67	56.75	45.10	19.10
Focus-dLLM (α=0.7)	17.71	29.36	23.61	22.52	19.12	26.97	21.58	76.50	90.78	41.00	96.67	66.85	56.72	45.34	19.14

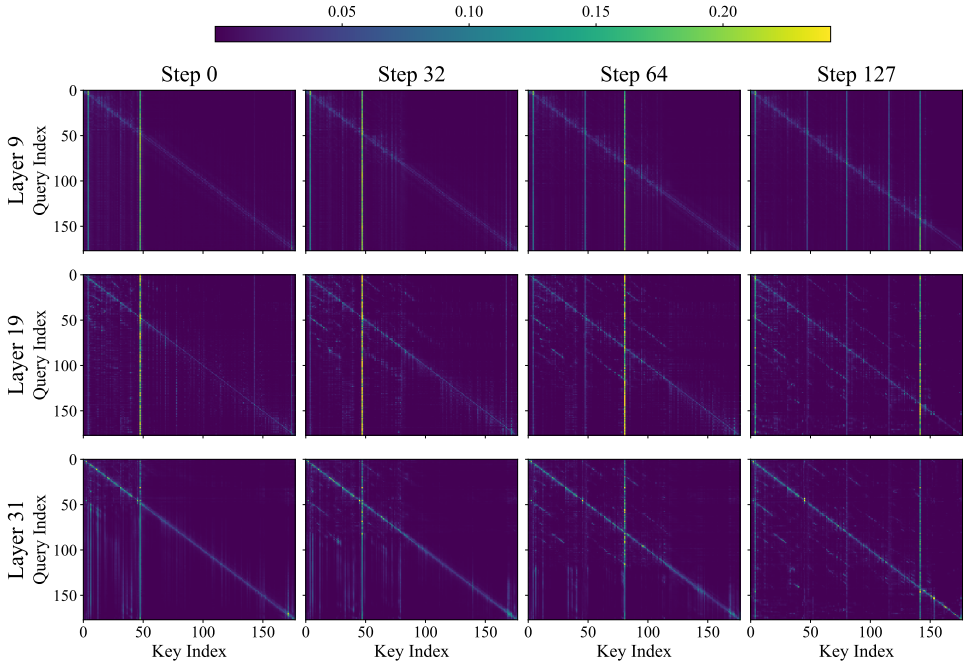

Figure 8: Attention patterns in LLaDA-8B-Instruct (Nie et al., 2025) across various layers and denoising steps. The heatmaps demonstrate the emergence of attention sinks (vertical bands) and their strong positional consistency across different layers within the same step.